REMARKS

This responds to the Office Action mailed on October 23, 2006, and the references cited therewith.

Claims 1, 6, 8, 12, 15 and 17 are amended, no claims are canceled, and no claims are added; as a result, claims 1, 3-8 and 11-18 are now pending in this application.

§103 Rejection of the Claims

Claims 1, 3, 5-8 and 11-18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Scott et al. (US 6,925,547: hereinafter 'Scott') in view of Fossum et al. (US 4,888,679: hereinafter 'Fossum').

Applicant respectfully submits that neither Scott nor Fossum, alone or in combination, teach or suggest a multiprocessor computer system having latency tolerant and scalable distributed shared memory as taught by Applicant and claimed in claims 1, 3, 5-8 and 11-18.

Scott describes a method for performing remote address translation in a multiprocessor system (Abstract, lines 1-2). Specifically, Scott describes an address translation method where a virtual address for a remote node is sent to the remote node and translated into a physical address at the remote node using a translation look-aside buffer (TLB) (*id.* lines 2-19).

Fossum describes a method and apparatus using a cache and main memory for both vector processing and scalar processing units (Abstract). Specifically, Fossum describes a scalar processor sending a vector load command to a vector processor, and also sending a vector prefetch request to the cache in response to a vector load instruction (col. 3, lines 17-20).

Neither Scott nor Fossum, however, alone or in combination, teach or suggest remote address translation as taught by Applicant and claimed in claims 1, 3, 5-8 and 11-18.

In the Office Action, the Examiner argues that Scott discloses "a Remote Address Translation table (RTT), wherein the RTT translates memory addresses received from other processing node such that the memory addresses are translated into physical addresses within the shared memory" (Office Action, p. 4, lines 3-13). As support of this, the Examiner points to Abstract, Figs. 4A, 4B, 5A & 5B and col. 25, lines 39-50.

Although the cited portions show that Scott translates remote virtual addresses to physical addresses at a remote node using a TLB at the remote node, Scott's TLB is different from the

RTT in Applicant's approach. As taught at page 5, lines 6-8 of "Remote Translation Mechanism for a Multi-node System" (U.S. Application No. 10/235,898), which is incorporated into Applicant's specification, Applicant's RTT is used only to translate virtual address references to a local node when that memory reference is coming from a remote node. A separate TLB associated with a local processor is used to translate virtual address references to the local node by the local processor. *See also id.* page 8, line 24 through page 9, line 23. Therefore, in Applicant's claimed approach, accesses from remote nodes don't have to fight local processes for space in the TLB. This reduces churning in the TLB.

In contrast to Applicant's separate translation approach for local and remote references, under Scott's approach, the TLB in a system HUB (SHUB) is used for both local and remote address references. Scott states that:

> (col. 14, lines 9-13) If the connection endpoint is the **local node**, then the address translation uses a **[external] TLB on the local SHUB**. However, if the connection endpoint is a remote node, the address translation uses a [external] TLB on the remote node.
>
> (col. 14, lines 47-52) The address translation mechanism used by CE 64 uses an external TLB located on the local SHUB, or an external TLB located on a remote SHUB, but **does not use the TLBs which are used by the processors** themselves to perform translation. Thus, the TLBs used by CE 64 may be referred to as "external" TLBs since they are external to the processors.
>
> (col. 18, line 65 through col. 19, line 4) Thus, the address translation mechanism may be used to perform both local and remote address translations, with the **[external] TLB on the local SHUB** used for translating a virtual address if a CD indicates that the **local node** is the connection endpoint, and the [external] TLB on a remote SHUB used for translating the virtual address if a CD indicates that the remote node is the endpoint.

Although Scott discloses that the processor's TLB is separate from the [external] TLB in the SHUB, the TLBs associated with the processors do not participate in translating memory references to the local node by the processors. Instead, as quoted above, the separate [external] TLB in the local SHUB translates memory references by the local processors to the local node as well as memory references to the local node received from other remote nodes. This is a different approach from the approach described by Applicant and claimed in claims 1, 8, 12 and 17.

In addition, the Examiner asserts that Scott teaches or suggests "the shared memory including a cache" (Office Action, p. 3, #6, lines 16-17). As support of this, the Examiner points to col. 16, lines 7-15 of Scott, part of which states that:

> **To support** local **address translation**, each **SHUB contains** a translation-lookaside buffer (TLB) 108 for performing local address translations for both block transfers and AMOs. A TLB is a cache that **holds only page table mappings**.

Applicant teaches, and claims in amended claims 1, 8, 12 and 17, using a processor cache to store data from recent memory references. This is different from the TLB described by Scott. Applicant has amended claims 1, 8, 12 and 17 to emphasize this difference.

For these reasons discussed above, neither Scott nor Fossum, alone or in combination, teach or suggest a multiprocessor computer system having latency tolerant and yet scalable shared memory as taught by Applicant and claimed in claims 1, 3, 5-8 and 11-18. Claims 1, 8, 12 and 17 have been amended to emphasize these differences.

With regard to claims 3, 11, 13 and 18, claims 3, 11, 13 and 18 are patentable as depending on a patentable base claim. In addition, neither Scott nor Fossum, alone or in combination, teach or suggest the shared memory having a plurality of cache coherence directories as taught by Applicant and claimed in claims 3, 11, 13 and 18.

In the Office Action, the Examiner asserts that Scott discloses the same limitation (p. 7, lines 9-15). As support of this, the Examiner points to col. 5, lines 47-67 of Scott, which partly states:

> ...In one embodiments, all of the coherence information is passed across the bus in the form of messages, and each processor on the bus "snoops" by monitoring the addresses on the bus and, if it finds the address of data within its own cache, invalidating that cache entry. Other cache coherence schemes can be used as well...

Applicant respectfully disagrees. Although the cited portion discloses use of a cache coherence method, the portion does not teach or suggest **using cache coherence directories located in the shared memory** to maintain cache coherence as described and claimed by Applicant. Applicant is unable to find such a teaching in any of the references considered by the Examiner.

With regard to claims 6 and 15, claims 6 and 15 are patentable as being dependent on a patentable base claim. In addition, neither Scott nor Fossum, alone or in combination, teach or suggest a scalar processing unit having a scalar cache memory with a subset of cache lines stored in a processor cache as taught by Applicant and claimed in claims 6 and 15.

In the Office Action (p. 7, line 20 through p. 8, line 2), the Examiner states that Fossum teaches the same limitation. As support of this, the Examiner points to Fig. 1 (CACHE 24) and col. 4, lines 15-54 of Fossum.

Applicant respectfully disagrees. Although the cited portions show use of a cache associated with both a scalar processor (21) and a vector processor (22), the portions do not show using an additional cache dedicated to the scalar processor as taught by Applicant and claimed in claims 6 and 15. Furthermore, Fossum does not teach or suggest specific way of connecting the scalar cache memory (920) to the cache (120) by having the scalar cache memory (920) contain only a subset of cache lines stored in the cache (120) as taught by Applicant and claimed in claims 6 and 15.

With regard to claim 4, claim 4 is patentable as being dependent on a patentable base claim. Reconsideration of claims 1, 3-8 and 11-18 is respectfully requested.

CONCLUSION

Applicant respectfully submits that the claims are in condition for allowance, and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicant's attorney at (612) 373-6909 to facilitate prosecution of this application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 19-0743.

Respectfully submitted,

STEVEN L. SCOTT

By his Representatives,

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. Box 2938
Minneapolis, MN 55402
(612) 373-6909

Date December 4, 2006

By Thomas F. Brennan
Thomas F. Brennan
Reg. No. 35,075

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being filed using the USPTO's electronic filing system EFS-Web, and is addressed to: Mail Stop Amendment, Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on this 4th day of December 2006.

CANDIS BUENDING

Name

Signature